IN THE CLAIMS

Claims 10-20 have been amended. All pending claims are reproduced below.

1-9 (Canceled).

10. (Currently Amended) A computer-implemented method for array shape inferencing comprising the steps of:

determining an input shape-tuple for each operand of a program expression of [an] a high-level array-based language wherein the size of at least one operand is unknown;

automatically analyzing the use of each operand in the program expression; and

determining prior to run-time a resulting shape-tuple of the program expression [using an algebraic framework].

11. (Currently Amended) The computer-implemented method of claim 10, wherein the high-level array based language is MATLAB.

12. (Currently Amended) The computer implemented method of claim 10, wherein determining prior to run-time a resulting shape-tuple of the program expression [using an algebraic framework] comprises the steps of:

determining a rank of the resulting shape-tuple; and,

promoting the input shape-tuple[s] for each operand to an appropriate rank.

13. (Currently Amended) The computer implemented method of claim 12, wherein determining the rank of the resulting shape-tuple comprises the steps of:

[determining] identifying a [the] rank of [each] the input [operand] shape-tuple for each operand;

identifying [an operator corresponding to the input operands] a built-in function in the program expression; and,

[ascertaining] determining the rank of the resulting shape-tuple according to the [operator] built-in function and the rank[s] of the input shape-tuple for each operand.

14. (Currently Amended) The computer-implemented method of claim 12, wherein promoting the input shape-tuple[s] for each operand to an appropriate rank comprises the steps of:

[identifying] comparing the rank of the resulting shape-tuple to the rank of the input shape-tuple for each operand;

responsive to the rank of the resulting shape-tuple being greater than the rank of the input shape-tuple for an operand, expanding the input shape-tuple[s] for the operand to correspond with the rank of the resulting shape-tuple; and,

appending trailing extents of the expanded input shape-tuple[s] for the operand with an appropriate value.

15. (Currently Amended) The computer-implemented method of claim [10] 13, further comprising the steps of:

[identifying a built-in function in the program expression;]

determining a shape-tuple operator for the built-in function; and,

applying the [operand] input shape-tuple[s] of each operand to the shape-tuple operator for the built-in function.

16. (Currently Amended) The computer-implemented method of claim 15, wherein determining a shape-tuple operator for the built-in function comprises the step of:

[identifying a shape-tuple expression corresponding to the built-in function; and[,]

assigning the shape-tuple expression as the shape-tuple operator.]

looking up, in a table, a shape-tuple operator corresponding to the built-in function.

17. (Currently Amended) The computer-implemented method of claim [16] 15, further comprising the step of [assigning] calculating a shape predicate [to] for the resulting shape-tuple.

18. (Currently Amended) The computer-implemented method of claim 10, further comprising the steps of:

performing an array conformability check at run-time for a first statement; and

applying a result of the conformability check to a second statement.

19. (Currently Amended) The computer-implemented method of claim 18, further comprising the step of:

determining a relationship among the first statement and the second statement.

20. (Currently Amended) [In the] The computer-implemented method of claim [18] 10, further comprising the step of:

preallocating [a] storage for each operand whose size is statistically unknown, based upon the input shape-tuple for each operand [shape to the variable of the statement in a loop execution].

21. (New) The computer-implemented method of claim 14, further comprising:

responsive to the rank of the resulting shape-tuple being less than the rank of the input shape-tuple for an operand, truncating the input shape-tuple[s] for the operand to correspond with the rank of the resulting shape-tuple.